

1

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-212807

(43)Date of publication of application : 06.08.1999

(51)Int.Cl.

G06F 9/455

G06F 9/45

(21)Application number : 10-018558

(71)Applicant : HITACHI LTD

(22)Date of filing : 30.01.1998

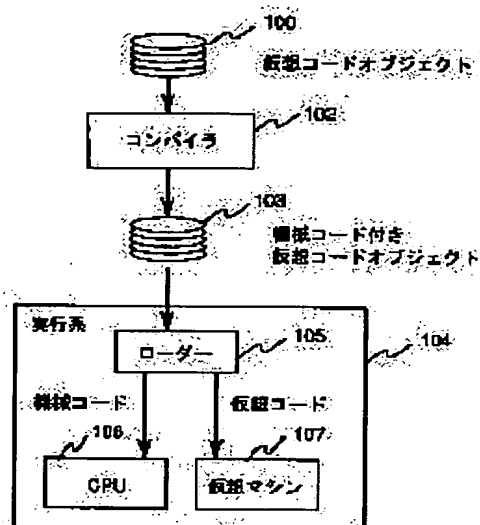
(72)Inventor : HIROSE TAKAHIRO

(54) PROGRAM EXECUTION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To improve the execution performance of a virtual computer system by compiling a virtual code into a machine language code, integrating the original virtual code and the generated machine language code into a single object file and executing the machine language code and the virtual code when the machine language code is adaptive and not adaptive to the architecture of an execution system respectively.

SOLUTION: A compiler 102 uses a virtual code object 100 as its input, compiles a virtual code to generate a machine code and outputs a virtual code object 103 having the machine code that is obtained by integrating both virtual and machine codes. An execution system 104 consists of a loader 105, a CPU 106 and a virtual machine 107. The loader 105 receives the virtual object having the machine code as its input, extracts the code from the object and distributes the processes to the CPU 106 and the machine 107 according to the type of the code. The CPU 106 and the machine 107 receive the machine code and the virtual code as their inputs and carry out the calculation processes respectively.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-212807

(43) 公開日 平成11年(1999) 8月6日

(51) Int.Cl.⁶

G 0 6 F 9/455
9/45

識別記号

F I

G 0 6 F 9/44

3 1 0 A

3 2 2 E

審査請求 未請求 請求項の数 3 O L (全 4 頁)

(21) 出願番号 特願平10-18558

(22) 出願日 平成10年(1998) 1月30日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 廣瀬 隆裕

神奈川県川崎市麻生区王禅寺1099番地株式

会社日立製作所システム開発研究所内

(74) 代理人 弁理士 小川 勝男

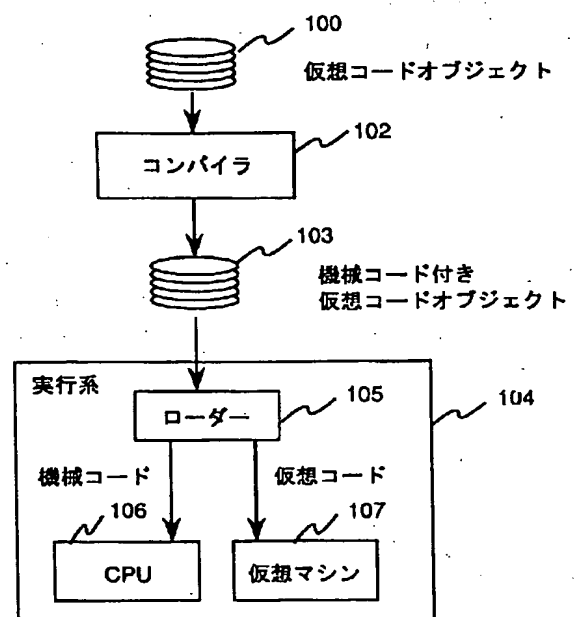
(54) 【発明の名称】 プログラム実行方法

(57) 【要約】

【課題】 仮想コードを実行する仮想計算機は、機械コードを実行する通常の計算機に比べて、実行性能が極端に低い。

【解決手段】 コンパイラにより、仮想コードから機械コードを生成し、両者を単一のオブジェクトファイルに統合する。実行時にオブジェクトを調べ、機械コードが実計算機に適合したら計算機で機械コードを実行し、適合しなかったら仮想機械で仮想コードを実行する。

図1



【特許請求の範囲】

【請求項1】プログラムのオブジェクトファイルにおいて、同一の手続きについて、実行マシンに依存する機械語オブジェクトコードと、実行マシンに依存しない仮想命令コードを格納し、実行環境にあわせて実行中でもコードを選択できるようにすることを特徴とするプログラム実行方法。

【請求項2】請求項1のプログラム実行方法において、実行マシンに依存する条件によって性能の向上が見込めない場合は、機械語コードをファイルから削除することにより、コードサイズの増加を抑えることを特徴とするプログラム実行方法。

【請求項3】プログラムのオブジェクトファイルにおいて、ファイルを識別するファイルヘッダーと仮想マシン向けの仮想コードフィールドとオプションフィールドをもち、オプションフィールド中に実行環境に依存する機械コードとコード種別情報を保持することを特徴とするファイル形式。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 計算機の利用において、プログラムを実行する方式に関する。

【0002】

【従来の技術】 計算機で実行するオブジェクトプログラムの形式は、計算機システムによって異なる。ある計算機向けに作られたオブジェクトプログラムは、別の計算機で実行することが出来ない。同じ仕事をさせるためであっても、計算機が異なればオブジェクトプログラムを作り直さなくてはならない。

【0003】 このような不便を避ける方法として、仮想計算機を定義する方法がある。仮想計算機の例として、JavaSoft: "The Java Virtual Machine Specification", Addison Wesley 等に記述がある。仮想計算機とは、仮想的な実行環境をソフトウェアで作り出す方法である。仮想計算機向けに作られたオブジェクトプログラムは、仮想計算機ソフトを移植すれば、任意の計算機で実行できる。

【0004】

【発明が解決しようとする課題】 仮想計算機はソフトウェアで実現されているため、オブジェクトプログラムをCPUで直接実行する方式に比べて、実行性能が極端に悪くなってしまう。

【0005】 実行性能を改善する方法として、JIT(Just In Time)コンパイラ方式が提案されている。実行系の中に、仮想コードを機械語コードに翻訳するコンパイラを組み込んでおく。仮想コードを受け取ったら、コンパイラで翻訳し、生成した機械語コードをCPUで直接実行する。

【0006】 実効速度は改善するが、実行系に大きな負荷がかかる。本来の処理とは別にコンパイラが動くの

で、メモリを大量に消費する。コンパイルに要するCPU負荷も大きい。

【0007】

【課題を解決するための手段】 (1) 仮想コードを機械語コードにコンパイルし、元の仮想コードと生成した機械語コードを単一のオブジェクトファイルに統合する。

【0008】 (2) 実行系は、上記のオブジェクトファイルをロードする。機械語コードが実行系のアーキテクチャに適合する場合は機械語コードを実行し、適合しない場合は仮想コードを実行する。

【0009】

【発明の実施の形態】 図1に全体構成を示す。コンパイラ102は、仮想コードオブジェクト100を入力として受け取り、仮想コードをコンパイルして機械コードを生成し、両者を統合した機械コード付き仮想コードオブジェクト103を出力する。

【0010】 実行系104は、オブジェクトを読みとるローダー104と、計算処理を行うCPU106と仮想マシン107から構成される。ローダー105は、機械コード付き仮想オブジェクトを入力として受け取り、オブジェクトの中からコード取り出し、コードの種類によって処理をCPU106と仮想マシン107に振り分ける。

【0011】 CPU106は、機械コードを入力として受け取り、計算処理を実行する。

【0012】 仮想マシン107は、仮想コードを入力として受け取り、計算処理を実行する。

【0013】 図2は、機械コード付き仮想コードオブジェクトの一例を示している。オブジェクトの形式は仮想マシンの定義によって異なる。ここでは、説明のために、かなり簡略化されたオブジェクト形式を取り上げている。機械コード付き仮想コードオブジェクト103は、ファイルヘッダー201と仮想コードフィールド202とオプションフィールド203から構成されている。ファイルヘッダー201には、ファイル種別や仮想コード種別のような情報が書き込まれている。仮想コードフィールド202には、仮想マシンが実際に処理する仮想コードが書き込まれている。

【0014】 オプションフィールド203は、文字どおりオプションであり、どのような情報が書かれるかは実装に依存する。実行系104は、自身が対応していないオプション情報に出会ったら、処理せずに読み飛ばす。一般には、ベンダー情報やデバッグ情報のようなものを書き込むことが多い。コンパイラ102は、オプションフィールド203に、コード種別情報204と機械コード205を書き込む。コード種別情報204は、機械コードのアーキテクチャ情報やバージョン情報を持っている。機械コード205は、仮想コードをコンパイルして得たコードである。

【0015】 図3は、コンパイラの処理を示している。仮想コード解析301は、仮想コードオブジェクトを入力として受け取り、解析を行い、解析結果を出力する。中

3

間コード生成302は、解析結果を受け取り、最適化に適した中間コードに変換して出力する。最適化303は、中間語を受け取り最適化を行う。機械コード生成304は、最適化済み中間コードを受け取り、機械コードに変換する。機械コード付き仮想コードファイル生成305は、生成した機械コードと、元の仮想コードを受け取り、単一のオブジェクトファイルに統合処理を行い、機械コード付き仮想コードオブジェクト103を出力する。

【0016】図4は、実行系の処理を示している。まず401で、機械コード付き仮想コードオブジェクト401を読み取る。402で、ファイルのオプションフィールドを調べる。実行系の対応していないフィールドは読み飛ばし、コード種別情報と機械コードを探す。403で、機械コードが見つからなければ処理を406に移し、見つかったら404に処理を移す。404に移ったら、コード種別情報を参照し、アーキテクチャやバージョンが実行系のCPUに適合するかどうか調べる。適合したなら405に処理を移し、適合しなかったら406に処理を移す。

【0017】405に移ったら、機械コードをシステムの

4

実行コード領域にロードする。ロードが終わったら、106に移り、ロードしたコードをCPUで直接実行する。

【0018】406に処理が移ったら、仮想コードを仮想マシンの実行コード領域にロードする。ロードが終わったら、107に移り、ロードしたコードを仮想マシンで実行する。

【0019】

【発明の効果】仮想計算機システムの実行性能を向上することができる。

10 【図面の簡単な説明】

【図1】プログラム方法の全体構成。

【図2】機械コード付き仮想コードオブジェクトの例。

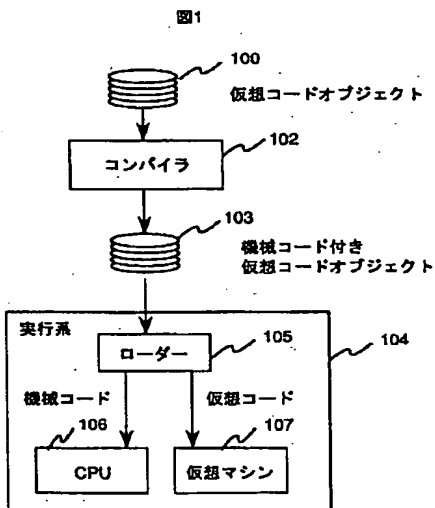
【図3】コンパイラの処理。

【図4】実行系の処理。

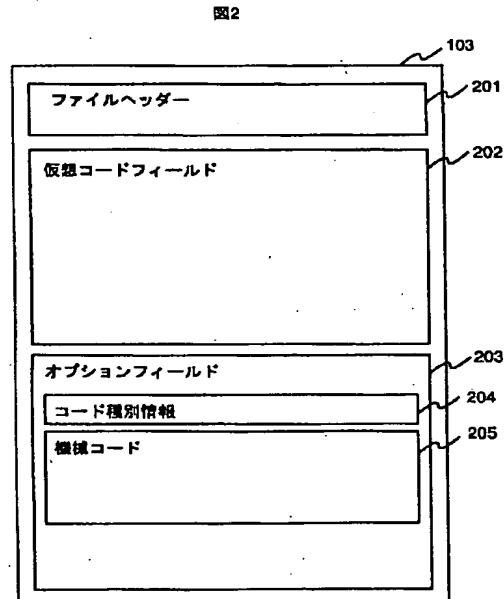
【符号の説明】

100…仮想コードオブジェクト、102…コンパイラ、103…機械コード付き仮想コードオブジェクト、104…実行系、105…ローダー、106…機械コード、107…仮想コード。

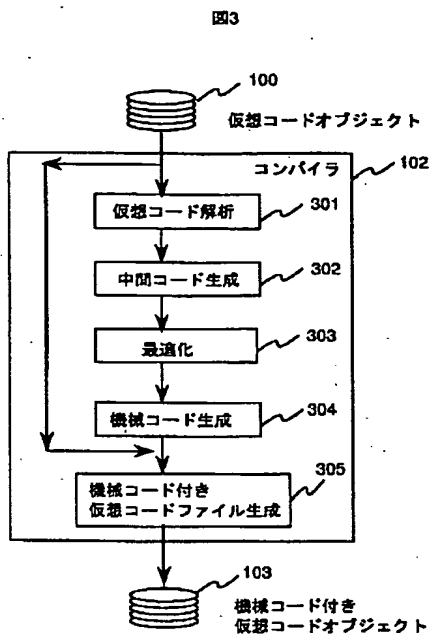
【図1】



【図2】



【図3】



【図4】

